High speed low complexity radix-16 Max-Log-MAP SISO decoder

Oscar David Sanchez Gonzalez, Christophe Jego, Michel Jezequel, Yannick Saouter

High Speed Low Complexity Radix-16 Max-Log-MAP SISO Decoder

Oscar Sánchez*, Christophe Jégo†, Michel Jézéquel*, Yannick Saouter*
*Institut MINES-TELECOM TELECOM Bretagne UMR, CNRS 6285 Lab-STICC
Technopôle Brest Iroise CS 83818, 29238 Brest Cedex 3, France
Email: {oscar.sanchez-gonzalez, michel.jezequel, yannick.saouter}@telecom-bretagne.eu
†IPB / ENSEIRB-MATMECA CNRS IMS, UMR 5218 351
Cours de la Libération 33405 Talence
Université de Bordeaux, France
Email: christophe.jego@enseirb-matmeca.fr

***Abstract*—At present, the main challenge for hardware implementation of turbo decoders is to achieve the high data rates required by current and future communication system standards. In order to address this challenge, a low complexity radix-16 SISO decoder for the Max-Log-MAP algorithm is proposed in this paper. Based on the elimination of parallel paths in the radix-16 trellis diagram, architectural solutions to reduce the hardware complexity of the different blocks of a SISO decoder are detailed. Moreover, two complementary techniques are introduced in order to overcome BER/FER performance degradation when turbo decoders based on the proposed SISO decoder are considered. Thus, a penalty lower than 0.05dB is observed for a 8 state binary turbo code with respect to a traditional radix-2 turbo decoder for 6 decoding iterations.**

***Index Terms*—Turbo codes, high radix architectures.**

I. Introduction

The adoption of turbo codes [1] in last generation wireless communication systems, such as 3GPP-LTE and WiMAX, has prompted research activities towards the design of high throughput Turbo Decoder (TD) architectures. In next generation standards, throughput requirements are significantly higher. For instance, LTE-advance standard targets data rates around 1Gbit/s. Thus, the design of low complexity high throughput TD architectures becomes a major challenge. The decoding of turbo codes is carried out through an iterative process where two Soft-Input Soft-Output (SISO) decoders, operating in natural and interleaved domain, exchange extrinsic information. The inherent recursive nature of MAP-based algorithms (Log-MAP or Max-Log-MAP [2]), implemented by each SISO decoder in the TD, is an important issue that limits the TD throughput. To overcome this constraint, radix-2^{N_T} architectures have been proposed [3]–[7]. In these architectures, N_T transitions in the trellis diagram are performed per clock cycle.

Most of the works in the literature that propose radix-2^{N_T} architectures are concerned to high throughput, and only a few explore the hardware complexity reduction. A radix-16 ACS unit to decompose the compare operation among 16 branches into two levels is presented in [6]. The hardware overhead is minimized thanks to a two-dimensional ACS unit architecture. In [7], a radix-16 modified log-MAP algorithm and a two-stage compare and select architecture to further decrease the latency and processing power is detailed. The main novelty of our work is the design of a radix-16 SISO architecture based on a radix-8 ACS unit for the Max-Log-MAP algorithm. Since the ACS unit is responsible of the bottleneck in the decoder, a reduction of the SISO critical path with respect to the architectures in [6] and [7] is possible. Thus, the SISO decoder throughput can be improved. Furthermore, a hardware complexity reduction of all the block units that form the SISO decoder is proposed. A clever idea is also introduced to avoid TD BER/FER performance degradation.

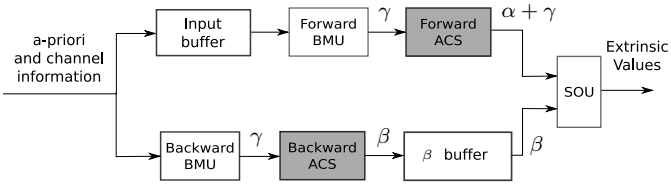

Fig. 1. SISO decoder architecture.

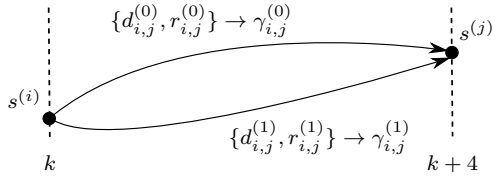

Fig. 2. Radix-16 ACS unit trellis diagram transition.

The rest of the paper is organized as follows. In section II the proposed radix-16 SISO decoder architecture is presented. Each block unit of the designed SISO decoder is detailed. Then, a comparison in terms of BER/FER performance of different versions of the turbo decoder is given in section III. Synthesis results are summarized in section IV. Finally, section V concludes the paper.

II. Proposed Radix-16 SISO architecture

Figure 1 details the block diagram of the considered SISO decoder. Inputs are the channel information − systematic $L(d_k)$ and redundant $L(r_k)$ LLR − and *a-priori* (L_k^a) values. The Branch Metric Units (BMU) calculate γ values used by the ACS units. The Soft Output Unit (SOU) computes extrinsic values L_k^e and takes hard decisions $\hat{d}_k$. To ensure that the critical path is in the ACS unit, pipeline stages can be included in the BMUs and SOU.

The 8-state binary turbo code compliant with the LTE standard has been selected to illustrate our propositions. Let $S = \{s^{(0)}, \ldots, s^{(7)}\}$ be the set of convolutional encoder states, and $\sum_{m=0}^{2} a_m^{(i)} \cdot 2^m$ the decimal representation of the state $s^{(i)}$, with $a_m^{(i)} \in \{0,1\}$ the value of the flip-flops that form the convolutional encoder. Consider Fig. 2 that depicts the state transition $s^{(i)} \rightarrow s^{(j)}$ in the trellis diagram of a radix-16 ACS unit ($N_T = 4$). For this radix value, two paths (parallel paths) are possible from any state at time k to any state at time $k+4$. Let these two paths be due to the systematic bit sequence $d_{i,j}^{(b)} = (d_k^{(i,j,b)}, d_{k+1}^{(i,j,b)}, d_{k+2}^{(i,j,b)}, d_{k+3}^{(i,j,b)})$ that produces the redundant bit sequence $r_{i,j}^{(b)} = (r_k^{(i,j,b)}, r_{k+1}^{(i,j,b)}, r_{k+2}^{(i,j,b)}, r_{k+3}^{(i,j,b)})$, with $b \in \{0,1\}$. Let $\gamma_{i,j}^{(b)}$ denote the branch metric value for the

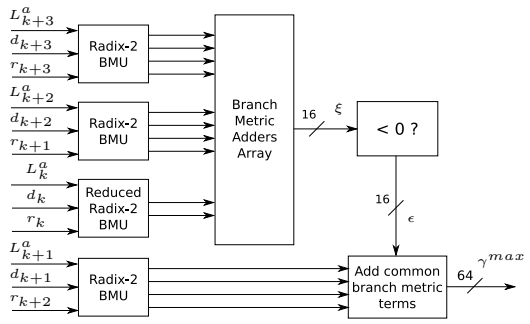

Fig. 3. Proposed radix-16 BMU architecture.

path b. As pointed out in [8], parallel paths should be eliminated prior to the ACS unit. Thus, it is possible to replace a radix-16 ACS unit by a less complex radix-8 ACS unit where each pair of states $(s^{(i)}, s^{(j)})$, at time k and $k+4$ respectively, are connected only by one path[1]. The branch metric value for this path is then $\gamma_{i,j}^{max} = \max(\gamma_{i,j}^{(0)}, \gamma_{i,j}^{(1)})$. This simplification does not affect the result of the ACS unit operation. Furthermore, since a radix-8 ACS unit has a lower critical path compared to a radix-16 ACS unit, it enables to increase the clock frequency of the corresponding decoder. The systematic and redundant bits that define the parallel paths in the transition $s^{(i)} \rightarrow s^{(j)}$ are given by:

$$\begin{aligned} d_{i,j}^{(b)} =& (d_k^{(i,j,b)}, d_{k+1}^{(i,j,b)}, d_{k+2}^{(i,j,b)}, d_{k+3}^{(i,j,b)}) \\ =& (a_0^{(i)} + a_1^{(i)} + b, a_1^{(i)} + a_2^{(i)} + a_0^{(j)}, \\ & a_2^{(i)} + a_1^{(j)} + b, a_0^{(j)} + a_2^{(j)} + b) \\ r_{i,j}^{(b)} =& (r_k^{(i,j,b)}, r_{k+1}^{(i,j,b)}, r_{k+2}^{(i,j,b)}, r_{k+3}^{(i,j,b)}) \\ =& (a_0^{(i)} + a_2^{(i)} + b, a_1^{(i)} + a_0^{(j)} + b, \\ & a_2^{(i)} + a_0^{(j)} + a_1^{(j)}, a_1^{(j)} + a_2^{(j)} + b) \end{aligned} \quad (1)$$

A. Branch Metrics Unit Architecture

From (1), for any pair of parallel paths, $d_{k+1}^{(i,j,0)} = d_{k+1}^{(i,j,1)}$ and $r_{k+2}^{(i,j,0)} = r_{k+2}^{(i,j,1)}$ because they are independent of b. Therefore, the channel values corresponding to the bits d_{k+1} and r_{k+2} do not affect the choice of the maximum branch metric in the radix-16 ACS unit transition $s^{(i)} \rightarrow s^{(j)}$. $\gamma_{i,j}^{max}$ can be then computed as follows. First, a partial term $\xi_{i,j}$, due to the bits that are different in both paths, for $b = 0$ is calculated. If $\xi_{i,j} < 0$ then $\gamma_{i,j}^{(1)}$ is the maximum branch metric and $\epsilon_{i,j} = -\xi_{i,j}$. Otherwise, the maximum branch metric is $\gamma_{i,j}^{(0)}$ and $\epsilon_{i,j} = \xi_{i,j}$. Afterwards, $\gamma_{i,j}^{max}$ is obtained by adding to $\epsilon_{i,j}$ the terms corresponding to the bits d_{k+1} and r_{k+2}. The proposed BMU architecture is given in Fig. 3. Note that only 16 values $\xi_{i,j}$ have to be computed for all the 64 state transitions $s^{(i)} \rightarrow s^{(j)}$, $i, j = 0, \ldots, 7$. This BMU only costs 53% of the hardware resources of a conventional implementation. A low complexity adder-sharing BMU is proposed in [7]. It requires a memory to store 128 branch metrics. With our BMU architecture this memory is reduced to the half (64 $\gamma_{i,j}^{max}$ values). Moreover, the BMU in [7] needs additional registers that increase the latency since more cycles are required to share the adders. In our architecture this disadvantage is overcome.

B. Radix-8 ACS Unit Architecture

Figure 4 details the proposed radix-8 ACS unit. This unit is composed of four radix-2 ACS units and one radix-4 Compare-Select (CS) unit. First, eight additions are performed (state metrics plus branch metrics). Then, four values are selected. These values are sent to the radix-4 CS that has to produce the final state metric value. In order to reduce the ACS unit critical path, we have applied the modulo normalization technique as presented in [9]. Thus, state metric normalization blocks have been removed. Besides, we have adopted the radix-4 CS architecture presented in [5]. Indeed, this architecture has been optimized in terms of throughput.

[1]Note that although a radix-8 ACS unit is used, the SISO decoder corresponds to a radix-16 architecture ($N_T = 4$).

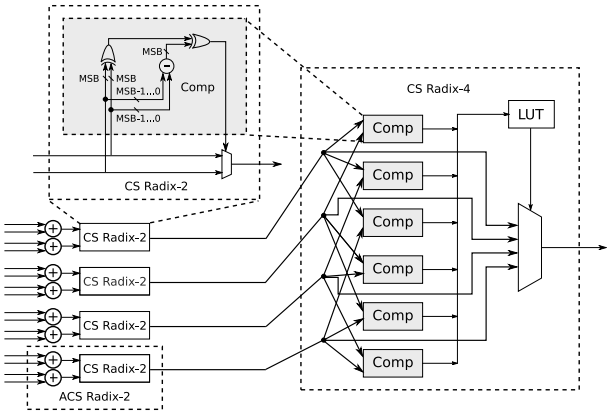

Fig. 4. Radix-8 ACS unit used.

C. Soft Output Unit Architecture

Since it is not possible to determine a priori which paths are eliminated in each state transition, a static comparison structure cannot be used for the SOU. Note however that we can group the 64 state transitions $s^{(i)} \rightarrow s^{(j)}$ in eight sets Q_l – each one having eight elements (pair of states) – for $l = 0, \ldots, 7$, such that $Q_l = \{(s^{(i)}, s^{(j)}) : d_{i,j}^{(0)} = (c_0^{(l)}, c_1^{(l)}, c_2^{(l)}, c_3^{(l)})\}$, with $c_n^{(l)} \in \{0, 1\}$, for $n = 0, 1, 2, 3$. Therefore, the eight branch metric values $\gamma_{i,j}^{max}$ with starting and ending states $s^{(i)}$ and $s^{(j)}$ respectively, such that $(s^{(i)}, s^{(j)}) \in Q_l$, correspond to only two possible systematic bit sequences, according to $d_{i,j}^{(b)}$ in (1). For instance, if $(c_0^{(0)}, c_1^{(0)}, c_2^{(0)}, c_3^{(0)}) = (0, 0, 0, 0)$, then the values of $\gamma_{i,j}^{max}$ related to the elements of C_0 are for the systematic bit sequence $(0, 0, 0, 0)$ or $(1, 0, 1, 1)$. Based on this observation we propose in Fig. 5 an original SOU architecture that contains comparing-routing elements. First, the 64 values $L = \alpha_k + \gamma^{max} + \beta_{k+4}$, corresponding to the non discarded paths, are computed. These values are then sent to eight *Max-L* blocks. The eight inputs of the l-th *Max-L* block correspond to the values L for the paths defined by the states $(s^{(i)}, s^{(j)}) \in Q_l$. Each *Max-L* block processes its inputs in three stages. In the first stage, four *Switch-I* blocks (Fig. 5(b)) either find in parallel the maximum values of their inputs or directly send the inputs to the outputs. In the two other stages, *Switch-II* blocks are used (Fig. 5(c)). These blocks find the maximum value or let one of the inputs to pass through them. Thus, at the output of the *Max-L* blocks the maximum value for each d is found[2]. These values are then compared using a fixed tree structure composed of *Switch-II* blocks, such that the number of comparators is minimized. Finally, the four extrinsic values are computed. Due to the elimination of paths, values L_{k+h}^e, for $h = 0, 2, 3$, may not be valid. In this case, a multiplexer enables to replace them by $p \cdot q_{k+h}$, where $q_{k+h} = \pm 1$, and $(q_{k+h}+1)/2$ is the hard decision corresponding to the systematic bit d_{k+h}. An expression for p will be proposed in Section III. On

[2]Note that it is possible that there is not a valid value for a specific d, since all its related paths could have been eliminated in the BMU unit.

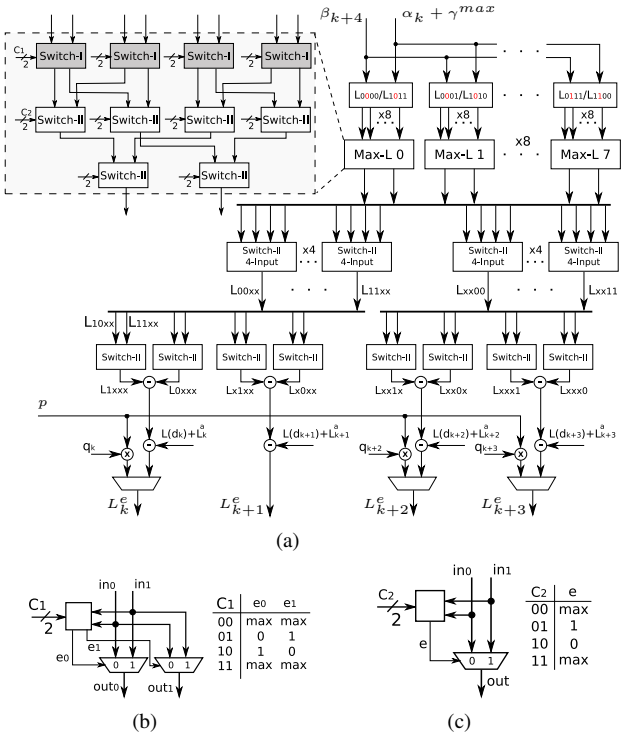

Fig. 5. (a) Proposed radix-16 SOU. (b) *Switch-I* circuit. (c) *Switch-II* circuit.

the other hand, the value L^e_{k+1} is always valid since the systematic bit d_{k+1} does not change in the parallel paths.

III. Turbo decoder performance comparison

The proposed radix-16 SISO architecture significantly affects the asymptotic gain of the TD. We have observed a penalty around 0.2dB, for code rates $R = 1/2$ and $1/3$, at FER of 10^{-6}, with an error floor that remains high compared to the error floor of a radix-2 SISO architecture. For high SNR values (error floor region), a-priori values grow fast during the iterative process, and thus, it is more probable to eliminate all the paths for a specific bit value. For instance, if the a-priori value for d_k (L^a_k) is high and positive, it is highly probable that all the paths corresponding to $d_k = 1$ are chosen, eliminating all the paths for $d_k = 0$. It means that L^e_k cannot be computed. Furthermore, this high a-priori value affects the computations of L^e_{k+2} and L^e_{k+3}, and thus, an undesirable correlation may appear between the hard decisions $\hat{d}_k$, $\hat{d}_{k+2}$ and $\hat{d}_{k+3}$ during the iterative process. To solve this problem, we propose two techniques: 1) select an appropriate value p when there is not enough information to compute a determinate extrinsic value (section III-A); 2) reduce the mutual interference between the extrinsic values in a same radix-16 trellis transition (section III-B).

A. Expression of p for Unknown Extrinsic Value

We have established an expression for p following a similar approach to the one presented in [10] for the decoding of Block Turbo Codes. In [10], when there is not competing codeword in order to compute the reliability value of a certain bit, the soft output is calculated as the sum of the magnitude of a set of channel observations at the input of the decoder. Since there are only four systematic bits for each radix-16 trellis transition, we have modified the summation by a minimum operation. Thus, we avoid too optimistic extrinsic values that are inconvenient during the iterative process performed by the turbo decoder. The value of p corresponds then to the minimum

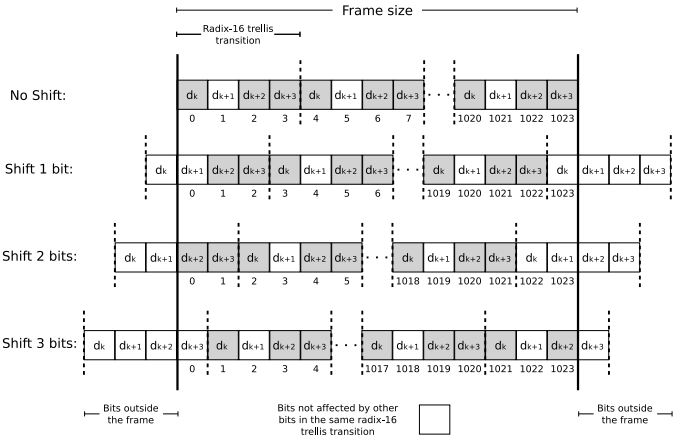

Fig. 6. SISO decoder displacement to avoid interference between symbols in a radix-16 trellis transition.

reliability value at the SISO decoder input (absolute value of the systematic plus the a-priori LLR values) between all the four bits in the radix-16 trellis transition as shown by (2). Extensive Monte-Carlo simulations showed the convenience of this expression.

$$p = \min_{i=0,1,2,3} \left(|L^a_{k+i} + L(d_{k+i})| \right) \tag{2}$$

B. Reduce the Interference of Bits in the Same Trellis Transition

We emphasize that the bit d_{k+1} affords a higher level of protection compared to the others bits in the radix-16 trellis transitions. Therefore, we propose to apply a *shift* operation on the frame treated by the SISO decoder, as presented in Fig. 6 for a frame size of 1024 bits. For this example, when no shifting is applied, bits $1, 5, 9, \ldots, 1021$ are protected. With a shift of one bit, the bits $0, 4, 8, \ldots, 1020, 1023$ are protected. With a shift of two or three bits, other bits are protected. Note also that the bits that interfere are different, in each radix-16 trellis transition, in function of the shift value. Thus, if we change the number of shifted bits in consecutive turbo decoder iterations, the negative effects introduced by our SISO architecture can be significantly reduced. This technique can be adapted to the sliding window process [11], and when multiples SISO decoders are assigned for the decoding of a frame [12].

If the shifting technique is applied, there are some bits that do not belong to the frame for the first and last radix-16 trellis transitions (Fig. 6). Therefore, the SISO decoder must be fed with appropriate a-priori values for these bits outside the frame, such that α and β values at the frame limits are not altered. Moreover, if the tail biting approach is used [13], bits outside the frame of the first (last) transition correspond to bits of the last (first) transition since the frame is circular. Thus, no special consideration should be given in this case.

BER and FER performance after six iterations for the LTE TD with 1024 bits per frame are given in Fig. 7[3]. Two architectures have been considered: one based on a radix-2 SISO decoder and another one based on the proposed radix-16 SISO decoder. Two different code rates $R = 1/3$ and $1/2$ have been studied. The code rate $R = 1/2$ is achieved by puncturing the original code. 6 and 9 bits have been chosen for the channel and extrinsic values representation, respectively. For the radix-2 SISO decoder, 10 bits are necessary for α and β metrics. For the radix-16 SISO decoder, 12 bits are

[3]These results have been obtained without applying the sliding window technique.

TABLE I
HARDWARE AREA IN TERMS OF EQUIVALENT 2-INPUT (NAND) GATE COUNT FOR DIFFERENT SISO DECODERS (INPUT AND β BUFFER NOT INCLUDED). CLOCK FREQUENCY IS SET TO 200MHZ.

	Radix-2	Radix-4	Radix-16 (radix-16 ACS unit)	Radix-16 (proposed)
BMU	635	3.2k	28.6k	15.3k
ACS (Area for all the 8 states)	3.4k ([9], 10 bits)	8.2k ([5], 11 bits)	34.1k ([6], 12 bits)	16.6k (12 bits)
SOU	2.2k	5.3k	23.5k	18.3k
Total estimated SISO area	10.5k	28.1k	148.9k	82.1k

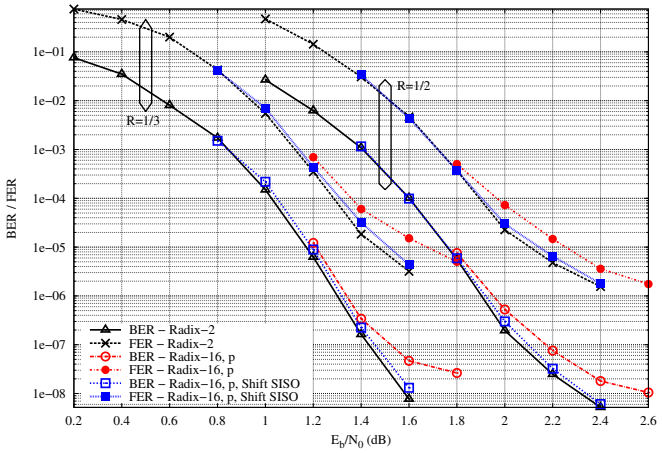

Fig. 7. Fixed point simulation for the LTE turbo decoder (1024 bits per frame), when radix-2 and the proposed radix-16 SISO are used.

necessary so that the modulo normalization technique can be applied. When p is given by (2), an error floor appears for both code rates. However, when the shifting technique is also applied, the architecture based on the proposed radix-16 SISO decoder exhibits a negligible degradation by comparison with an architecture based on the radix-2 SISO decoder.

IV. SYNTHESIS RESULTS

The proposed radix-16 SISO architecture was synthesized with a STMicrolectronics 90nm CMOS process ASIC target. The hardware complexity in terms of equivalent 2-input (NAND) gate count is given in Table I. In order to compare the SISO decoder hardware complexity as function of N_T, radix-2, radix-4 and radix-16 SISO decoders were also designed based on the scheme presented in Fig. 1. For the radix-2 and radix-4 SISO decoders, the ACS units in [9] and [5] were adopted, respectively. For the radix-16 SISO decoder, the two-dimensional ACS unit in [6] was chosen, and a SOU implementing the static comparison tree that minimizes the number of comparisons as presented in [7] was implemented. All SISO decoder architectures were synthesized for a 200MHz clock frequency. The input and β buffer complexities are not included in the comparison since their values depend on the window size. However, a radix-2^{N_T} ACS unit enables to reduce the β buffer size by a factor of N_T.

Thanks to the complexity reduction performed in all the SISO decoder blocks, our radix-16 SISO decoder is about 55% less complex than the considered radix-16 implementation. The proposed ACS unit helps considerably to achieve this result. Furthermore, the proposed BMU and SOU enable to overcome the hardware penalty cost introduced by a radix-16 approach. Compared to radix-2 and radix-4 SISO decoders, the proposed architecture is about 7.8 and 2.9 more complex, respectively. With this additional hardware complexity, the proposed SISO decoder improves by a factor of 4 and 2 the radix-2 and radix-4 SISO decoder throughput, respectively.

V. CONCLUSION

A low complexity radix-16 SISO decoder for the Max-Log-MAP algorithm is presented in this paper. Two complementary techniques have been proposed in order to limit the BER/FER performance degradation introduced by a TD architecture based on the proposed radix-16 SISO decoder. Moreover, an elimination of parallel paths in the trellis diagram allows us to use a radix-8 architecture for the ACS unit. The association of these different contributions enable to design a high speed low complexity radix-16 SISO decoder. The ideas presented in this paper can be applied to design higher radix SISO decoders with an acceptable cost-throughput ratio.

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes," in *IEEE ICC '93, Geneva*, 1993, pp. 1064 – 1070.

[2] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain," in *Communications, 1995. ICC '95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on*, vol. 2, jun 1995, pp. 1009 – 1013 vol.2.

[3] M. Bickerstaff, L. Davis, C. Thomas, D. Garrett, and C. Nicol, "A 24mb/s radix-4 logMAP turbo decoder for 3GPP-HSDPA mobile wireless," in *Solid-State Circuits Conference, 2003. Digest of Technical Papers. ISSCC. 2003 IEEE International*, 2003, pp. 150 – 484 vol.1.

[4] Z. Wang, "High-speed recursion architectures for MAP-based turbo decoders," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol. 15, no. 4, pp. 470 –474, april 2007.

[5] C. Studer, C. Benkeser, S. Belfanti, and Q. Huang, "Design and implementation of a parallel turbo-decoder ASIC for 3GPP-LTE," *Solid-State Circuits, IEEE Journal of*, vol. 46, no. 1, pp. 8 –17, jan. 2011.

[6] C.-H. Tang, C.-C. Wong, C.-L. Chen, C.-C. Lin, and H.-C. Chang, "A 952ms/s Max-Log MAP decoder chip using radix-4 x 4 ACS architecture," in *Solid-State Circuits Conference, 2006. ASSCC 2006. IEEE Asian*, nov. 2006, pp. 79 –82.

[7] K.-T. Shr, Y.-C. Chang, C.-Y. Lin, and Y.-H. Huang, "A 6.6pj/bit/iter radix-16 modified log-MAP decoder using two-stage ACS architecture," in *Solid State Circuits Conference (A-SSCC), 2011 IEEE Asian*, nov. 2011, pp. 313 –316.

[8] G. Fettweis and H. Meyr, "Parallel Viterbi algorithm implementation: breaking the ACS-bottleneck," *Communications, IEEE Transactions on*, vol. 37, no. 8, pp. 785 –790, aug 1989.

[9] C. Benkeser, A. Burg, T. Cupaiuolo, and Q. Huang, "Design and optimization of an HSDPA turbo decoder ASIC," *Solid-State Circuits, IEEE Journal of*, vol. 44, no. 1, pp. 98 –106, jan. 2009.

[10] P. Adde and R. Pyndiah, "Recent simplifications and improvements in Block Turbo Codes," in *2nd International Symposium on Turbo Codes & Related Topics, September 4-7, Brest, France*, 2000, pp. 133 – 136.

[11] S. Benedetto, G. Montorsi, D. Divsalar, and F. Pollara, "A Soft-Input Soft-Output Maximum A Posteriori (MAP) Module to Decode Parallel and Serial Concatenated Codes," *Telecommunications and Data Acquisition Progress Report*, vol. 127, pp. 1–20, Jul. 1996.

[12] Y. Zhang and K. Parhi, "Parallel turbo decoding," in *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on*, vol. 2, may 2004, pp. II – 509–12 Vol.2.

[13] C. Berrou, C. Douillard, and M. Jezequel, "Multiple parallel concatenation of circular recursive convolutional (CRSC) code," *Ann. Telecomm*, no. 3-4, pp. 166–172, March-April 1999.